To appear in the Proceedings of International Conference on Information Technology (ITCC 2004), Special Session on Networked Physical Security and Surveillance Las Vegas, NV, USA, April 5 -7, 2004

Distributed Omni-Video Arrays and Digital Tele-Viewer for Customized Viewing, Event Detection and Notification

Mohan M. Trivedi, Kohsia S. Huang, Tarak Gandhi, Brett Hall, and Kimberly Harlow
Computer Vision and Robotics Research (CVRR) Laboratory
University of California, San Diego
http://cvrr.ucsd.edu/

Abstract

Recent innovations in real-time machine vision, distributed computing, software architectures, and encrypted high-speed networking are expanding the available technology for intelligent camera arrays on televiewing and interactive applications. In this paper we describe research aimed at the realization of a powerful televiewing system using arrays of omnidirectional cameras. The video arrays provide an interactive, real-time, multi-resolution televiewing interface to multiple people including physical security and emergency response crews simultaneously. Security measures such as authentication and network encryption are implemented on multiple platforms. Fusion of high-resolution rectilinear image over the omnidirectional image is also carried out. Computer vision techniques based on motion analysis are used for detecting interesting events. The televiewing system is verified for the performance as well as physically tested on the Super Bowl 2003 event for crowd detection and traffic monitoring.

1. Introduction

Televiewing is a powerful tool for transmitting a visualization of a remote environment, a process that is important for many applications [1][2][3]. An urban traffic environment is a good example where televiewing is particularly powerful and useful [4][5][6]. As a solution to the need for faster response times to highway incident detection, using helicopters or planes is impractical and expensive. Inductive loop sensors inherently cannot detect a variety of traffic incidents. Our research here offers an innovative solution to these challenges through televiewing on omni-video networks. The motivation for our research was to realize a powerful and integrated incident detection and monitoring system. Automated software developed by our team will provide an automatic analysis of the current traffic conditions using clusters of cameras placed along the roadway. This analysis data will provide the emergency response team with enhanced information in order to detect and analyze a potential incident, observe the event, and get prepared for the situation. This system also has the potential to make travel safer, smoother, and more economical, reducing wasted fuel and pollution [3].

2. Digital Televiewing

Digital tele-viewer (DTV) uses omnidirectional vision sensors (ODVS) as its video sources. An ODVS uses a hyperboloid mirror to reflect an entire 360-degree view into the lens of a single camera, as shown in Figure 1. As compared to the mosaic of multiple perspective videos into a single panoramic view, ODVS allows for production of the same view range with only one camera, thus saving cost and communication bandwidth significantly. The client side can then unwarp the omnidirectional video for tailored perspective views by a nonlinear transform.

Figure 1. An omni-directional vision sensor (ODVS) and the camera image.

Figure 2. DTV technologies: DTV core, Secure DTV (S-DTV), and Portable DTV (P-DTV).

Besides efficient bandwidth, the most note-worthy advantage of DTV is that multiple clients can access the same ODVS through network simultaneously, and each one can choose their interested view by specifying an electronic pan-tilt-zoom (PTZ) setting. The cost of these advantages of DTV is lower image resolution because a larger view is covered by the same number of pixels, yet this can be fixed by using a high resolution camera. It can also be fixed by fusion the high-resolution image with the omnidirectional image, as will be discussed later.

Technology	Features	
DTV	• Cost efficient • General public access	• Simultaneous multi-user tailored televiewing
P-DTV	• Mobile and multi-platform • Portable/Cross-platform	• General public access
S-DTV	• Secure links • Encrypted networking • User authentication • User classification on security layers	• View blocking filters • User access log and usage archive

Technology	Hardware		Software	
	Server	Client	Server	Client
DTV	Axis Video Server	Desktop, Laptop	N/A	C++
P-DTV	Axis Video Server	Desktop, Laptop, Tablet PC, PDA	N/A	Java
S-DTV	Desktop + Capture Card	Desktop, Laptop	Java, Linux	Java

Table 1. Summarization of the generations of DTV technology.

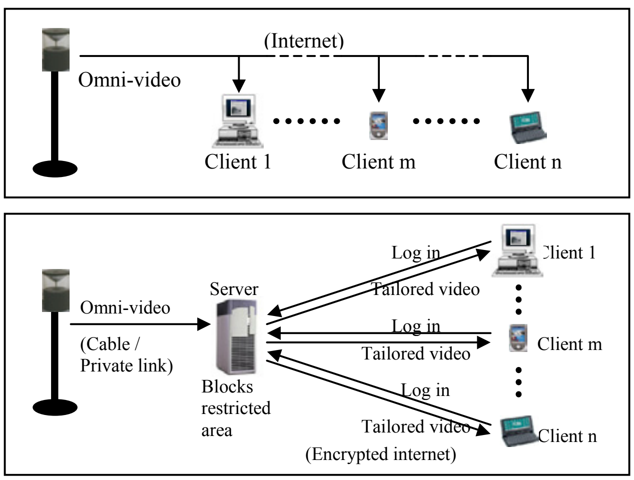

Figure 3. DTV and S-DTV system layout.

The variations of DTV generations are illustrated in Figure 2. The ODVS and perspective unwarping is the core of DTV technologies. For Portable DTV, the DTV is enhanced on the portability to operate on wireless-linked PDAs, Tablet PCs, and Laptops. The client software of P-DTV is Java-based to operate cross-platform. For Secure DTV, we emphasize the network security and accessibility features. The client software is Java-based to work cross-platform, yet for encryption it cannot stretch to PDAs due to their lack of encryption capabilities on currently available products. The features and implementation platforms of the generations of DTV are summarized in Table 1.

The layouts of DTV variations are compared in Figure 3. The DTV server uses MJPEG compression and sends the video stream to the clients via cable or wireless internet. The internet link may be public for P-DTV or encrypted for S-DTV. For S-DTV, the security is enhanced by physical security of the camera-server link, authentication of the users, classification of their security layers, blocking of their viewing according to the security layers, and archive of their usage.

A. DTV Core: ODVS Imaging and Perspective Selection

Our omni-camera is based on hyperboloid mirror. Hyperboloidal mirrors are found to be feasible for larger field of view [7] from a single viewing point [8]. By its optical property, the scene around the mirror is reflected into the camera sitting at the second focus of the mirror. Using inverse mapping, a panorama can be derived by back-projecting the omnidirectional camera image onto a virtual cylindrical screen around the mirror focus. Meanwhile, a perspective view can be derived by back-projecting the camera image onto a planar virtual screen whose orientation and distance to the mirror focus are specified by the pan, tilt, and zoom values, as detailed in [9]. The correspondence of the screen points to the camera image pixels can be stored in a look-up-table for faster view generation.

B. Multi-Platform Portable DTV

The DTV can be designed for multiple platforms include workstations, PCs, notebook PCs, Tablet PCs, and PDAs. The main advantage is on the portability of the DTV over all platforms supporting Java virtual machine (JVM), either wire or wireless linked. Potential users span from general public to police and first responders.

Java is selected as the implementation language for the P-DTV. Although Java has noticeably fewer available resources and is somewhat slower for real-time applications as compared to C or C++, however its portability outweighs these disadvantages. As processing speed in handheld devices increases, its cross-platform capability will continue to be an essential advantage.

For *handheld devices*, we evaluated their compatibility of Java virtual machine (JVM) libraries. Jeode, the JVM of Compaq iPAQ, has a very limited Java compatibility. Also the lower processing power of iPAQ limits the frame rate of the video stream. We also evaluated Sharp Zaurus, SonicBlue, and TabletPC. The basic structure of the DTV software is therefore adapted for the limitations of their JVM libraries as well as for the processing power such as the 206-MHz StrongARM and 400-MHz Transmeta TM3200 processors.

The basic *code structure* of the P-DTV as in Figure 4 has several layers. The lowest layer of the DTV structure is the parser, which decodes the MJPEG stream it receives into a pixel array. On the vision loop layer, the perspective view for a specific pan-tilt-zoom setting is generated from the pixel array by a look-up-table of the perspective transform. The table is recalculated only when a new camera or a new view is specified. The top layer runs the GUI frame thread of the DTV window. It takes the user settings from the GUI and switches to the specified camera view. The perspective video is also displayed at this layer.

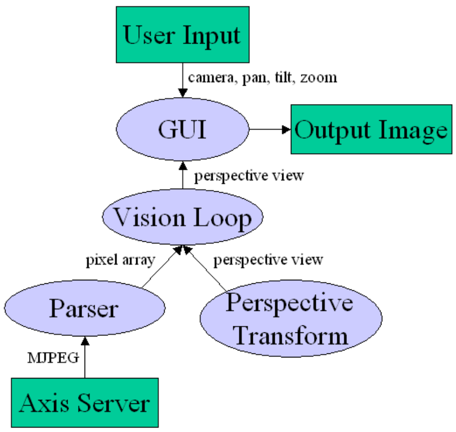

Figure 4. DTV application code flowchart.

The speed of JPEG decompression is an issue of the P-DTV due to the JVM libraries of the handheld devices as well as the processors. The frame rate can be increased by making the JPEG decompression more efficient. In addition, we have improved numerous limitations of the processing speed of the handhelds in the Java code. Also note that for P-DTV the Axis server is non-secure and not encrypted, the end user could use internet browser to view the omni-video. However to view the derived perspective view, one must use the P-DTV software.

C. Secure DTV

The S-DTV server is enhanced with five measures on hardware and software security: physical security, authentication, encryption, view blocking, and access archiving.

Physical Security: The link from the camera to the S-DTV server is a physically secured video cable. As shown in Figure 5, the S-DTV hardware is equipped with a capture card to take the video directly. In this case the S-DTV server is deployed with the ODVS at the same site, for example, at a lamp pole where the ODVS sits above the pole and the S-DTV server is locked in the base of the lamp pole, and the video cable is secured in the pole. For distributed omnicam arrays, private network is deployed to maximize the re-configurability, however, additional cost is involved to deploy the private network. The S-DTV software implementation also allows the combined configuration.

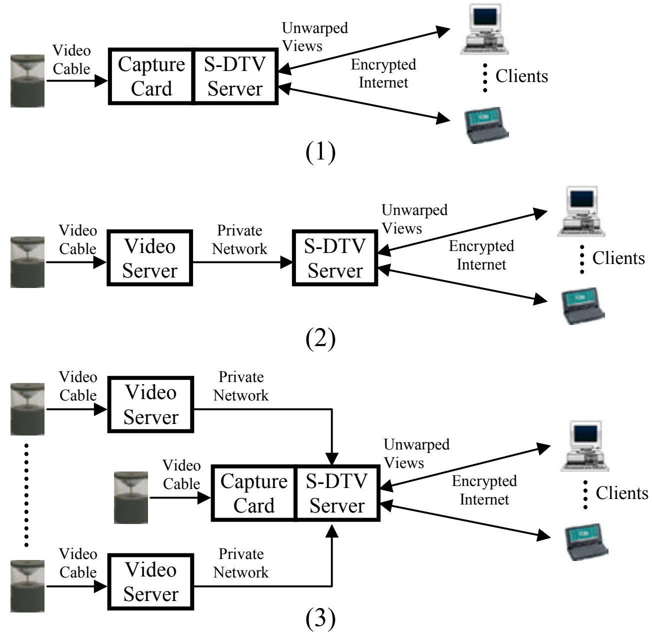

Figure 5. Direct video link, distributed video link, and hybrid configurations of the S-DTV server.

Authentication: Users need to have their accounts and passwords to access the S-DTV server through the encrypted internet link.

Encryption: 128-bit encryption secure links are connected between the S-DTV server and the clients. The Java Secure Socket Extension (JSSE) libraries, a standard library in Sun Java SDK 1.4, were used on major platforms. However, PDA Java versions only implement a portion of the standard Java libraries, and the S-DTV client currently will not run on any known PDA.

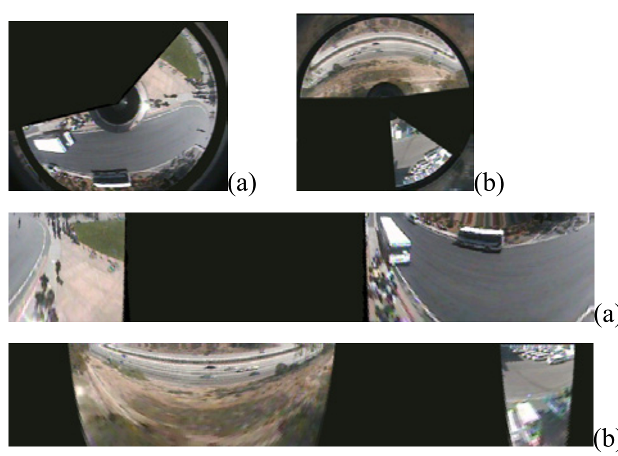

Figure 6. View blocking mechanism for lower security layer clients.

View Blocking: The secure-server version implements two levels of permission control: allowed camera and possible view angles of a particular camera. The server blocks out portions of the video stream for any areas outside that client's permissions and records the time and view that each client requests. Arbitrarily blockage shapes of black or red color can be overlaid on the original video using the Graphics2D library provided with the Sun Java SDK 1.4. Multiple blockage zones are defined according to various security clearance levels of the clients. Any specific user account may be enforced a different combination of the blockages or no blockage at all by the S-DTV server, as shown in Figure 6.

Access Archiving: All the transactions between the clients and the S-DTV server can be archived for later investigation and tracing.

3. Event Detection and Notification

Events can be extracted and summarized from the array videos of ODVS and high-resolution PTZ cameras by detecting and tracking features over frames. In particular, we identify events such as stalled vehicles, speeding vehicles, persons or vehicles in unusual places. Motion is an important cue in separating interesting events from extraneous clutter in the image. In this section, we briefly describe the approaches for motion-based event detection on stationary as well as mobile platforms.

Figure 7: Tracking of a vehicle in omni video sequence. Bounding box is shown in red, with track id in green, and the track history as a yellow line. Image on right shows the current estimate of the background image. Note that the moving vehicle has got removed in the background image.

Figure 8: Camera handover between an omnidirectional and a rectilinear camera. The numbers are track IDs in a single camera and letters are IDs assigned across cameras. A bus tracked in the omni camera is identified in the rectilinear camera by assigning the same letter (C).

Event Detection using Motion Cues: On stationary platforms, background subtraction is widely used to separate foreground objects [10]. By removing background from the current frame, foreground blobs are detected after morphological operations. We form the background by a running average of every pixel over the frames. The identified blobs are tracked over frames using Kalman filters, as shown in Figure 7.

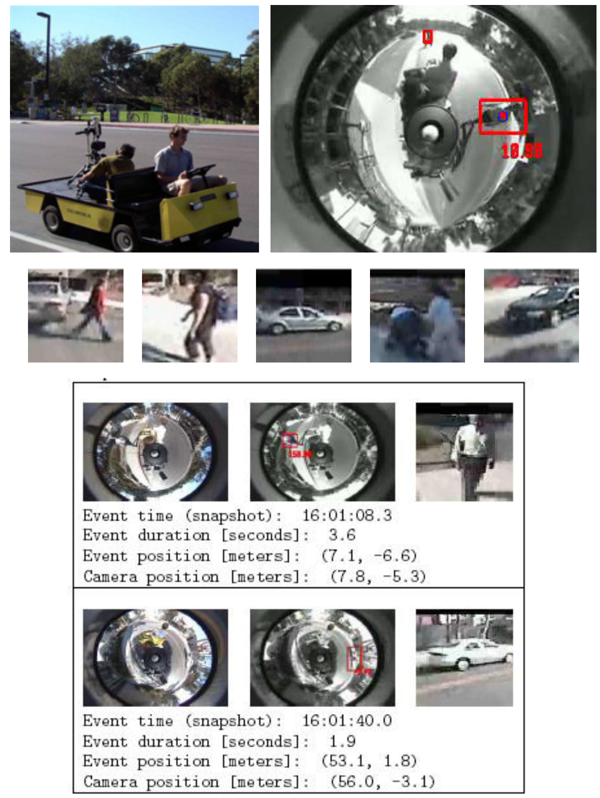

Figure 9: Event detection with a "mobile sentry" testbed. The ego-motion of ground plane is compensated and objects with independent motion are detected. Snapshots of interesting events are shown. The events can then be summarized in a tabular form.

Camera Handover: A single camera may not be sufficient to capture the entire area to be monitored. Hence, an array of cameras should be used. For such arrays, the event tracks leaving the field of view of one camera should be re-identified when they enter the other camera. For this purpose, a track is recorded in a handover table and assigned a letter code. The new track detected in another camera is matched with the tracks in the handover table. The matching is done by comparing the time of entry and track speed. Other features including color and size can also be used to improve the matching. Figure 8 is an example of camera handover.

Mobile Omni-Based Event Detection: On mobile platforms, background subtraction methods would fail due to camera motion. To detect foreground objects, ego-motion of camera should be compensated. The ground is approximated by a planar surface, whose ego-motion is modeled using a parametric transform [11][12]. We have generalized this model by combining the transformations converting the omni image to perspective view with the projective motion transform [13]. Iterative, coarse to fine, gradient based estimation is then used to estimate the motion parameters. Thus the foreground object is found as independent motion or height. The objects are tracked over frames, and the tracks persisting over a minimum number of frames are marked as interesting events, as shown in Figure 9.

4. Televiewing Testbed and Experimental Validation

In designing a televiewing testbed, some strategy is needed to obtain a good video coverage of the area [14]. We will focus on the combination of the ODVS and PTZ rectilinear cameras, which would enhance the available video by fusion and achieve robustness beyond what one type of camera alone can do. Although ODVS is sufficient for general purpose televiewing almost all the time, in resolution-demanding cases a PTZ camera is needed. On the other hand, since PTZ camera rotates with mechanical mount, important events could be missed when used on wide area surveillance.

Figure 10. UCSD base node capable of acquiring 16 video streams over a 1 Gigabit fiber optic link.

The outdoor camera clusters are located on streetlights beside I-5 freeway as well as near an intersection on campus where buses, cars, bikes, and people regularly pass. These are excellent locations under a variety of conditions. Each sensor cluster contains an ODVS and a PTZ rectilinear camera mounted in a weatherproof housing, as pictured in Figure 10. All of these camera videos are streamed directly back to the S-DTV server using fiber optics, a one Gigabit Cisco switch, and some Axis-2400 video server units (CVRR-Axis_'s). These video links are non-secure due to the Axis servers. However, the indoor camera is physically connected to the S-DTV server through video cable. Thus the S-DTV server is operating in the hybrid mode of Figure 5(3).

Figure 11: Results of seamlessly integrating a high resolution image to an omni-image. The GUI allows an observer to explore remote sites.

Figure 11 is an example showing the progress and results from the integration of a high-resolution rectilinear image with an ODVS view. This software enables the user to look around the area with the *virtual* PTZ of an ODVS. Using this interface, the user can view the high-resolution images in context of the background, creating a more immersive televiewing experience.

DTV Performance: The computation platforms involved in the experiment are listed as follows:

- *P-DTV Server*: Axis-2400 video server.
- *S-DTV Server*: Athlon XP 2400+ with 512 MB RAM running Linux RedHat 8.
- *Client 1*: P3 1.13 GHz with 768 MB RAM running Windows XP.
- *Client 2*: Tablet PC with 64 MB RAM running Windows 98.
- *Camera*: Sony EVI-370 with omnidirectional mirror.
- *Frame Grabber*: Hauppauge WinTV, installed in the S-DTV server.

	Operating System	Processor	Compression @ 0	Compression @ 40	Compression @ 50
Desktop	Windows XP	864-MHZ Pentium III	10 fps	19 fps	20 fps
Compaq iPAQ	Windows CE	206-MHz StrongARM	0.9 fps	1.2 fps	1.2 fps
Sharp Zaurus	Linux	206-MHz StrongARM		1.5 fps	
SonicBlue TabletPC	Windows 98	400-MHz Transmeta		2-3 fps	

Table 2. Integrated P-DTV client performance. Source image size is 352 by 240.

For S-DTV, frame rates of 3-4 fps with view blocking and 128-bit encryption were observed while running the client on the Tablet PC with 802.11 wireless ethernet. On notebook computers with wireless ethernet 11-12 fps was observed, and the frame rate increased to 14-15 fps using wired ethernet. For P-DTV, the performance varies widely with platforms as listed in Table 2. Examples of mobile P-DTV are shown in Figure 12.

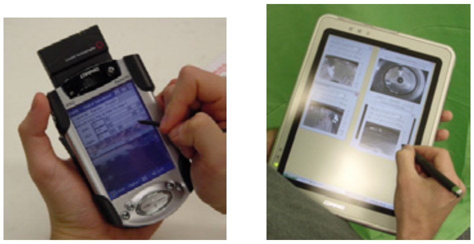

Figure 12. Mobile P-DTV on PDA and Tablet PC.

The video array was physically utilized on the Super Bowl 2003 event in San Diego, California for security monitoring. The cameras were wirelessly hooked up to the internet to monitor the southern perimeter of QUALCOMM Stadium, a high-traffic intersection in downtown San Diego called Gaslamp, and a waterfront command center staffed by San Diego Police and other security personnel. The camera networks were used to monitor traffic activity on the road in front of the Qualcomm Stadium using a vision-based traffic analysis and notification software. The program used median background subtraction to count cars and vehicle speeds. Also a dynamic crowd status analyzer and recorder was developed on the Gaslamp omnicam video, as shown in Figure 13. The software module used dynamic background subtraction and unwarped image statistics to estimate the size of the crowd gathered in the secured "party" area of the district, as well as the flow of pedestrians on the streets on a 24 hour basis. The numbers were updated every two minutes in a database for query and analysis. These two autonomous modules matched the requirements and purposes of the security agents in the specified public areas, proving an attractive future of the DTV technologies.

Figure 13. ODVS setup in downtown San Diego and crowd analysis for the Super Bowl 2003.

Acknowledgements

Our research is supported in part by the Center for Communication of Advanced Technology (CCAT), the von Liebig Center, and the Technical Support Working Group (TSWG). We wish to thank our colleagues from the CVRR lab who are also involved in related research activities. We would specifically like to thank Tasha C. Vanesian, Remy Chang, Greg Kogut, Dev Bala, and Rick Capella for their contributions.

References

[1] S. K. Nayar, T. Boult, "Omnidirectional Vision Systems: 1998 PI Report," *Proc. 1997 DARPA Image Understanding Workshop*, May 1997.

[2] T. E. Boult, R. J. Micheals, X. Gao, M. Eckmann, "Into the Woods: Visual Surveillance of Non-Cooperative and Camouflaged Targets in Complex Outdoor Settings," *Proc. IEEE*, vol. 89, no. 10, pp.1382-1402, Oct. 2001.

[3] K. Ng, H. Ishiguro, M. Trivedi, T. Sogo, "Monitoring Dynamically Changing Environments by Ubiquitous Vision System," *IEEE Visual Surveillance Workshop*, Fort Collins, Colorado, June 1999.

[4] M. Trivedi, I. Mikic, G. Kogut. "Distributed Video Networks for Incident Detection and Management," *Proc. 3rd IEEE Conference on Intelligent Transportation Systems*, Oct. 2000.

[5] B. Stewart, I. Reading, M. Thomson, C. Wan, and T. Binnie, "Directing Attention for Traffic Scene Analysis," P*roc. of IEE Int'l Conf. on Image Processing and its Applications (IPA '95)*, pp. 801-805, 1995.

[6] J. Miura, T. Kanda, and Y. Shirail, "An Active Vision System for Real-Time Traffic Sign Recognition," *Proc. IEEE Int'l. Conf. on Intelligent Transportation System (ITSC '00)*, pp. 52-57, Oct. 2000.

[7] S. Baker and S. Nayar, "A Theory of Catadioptric Image Formation," *Proc. 6th Int'l. Conf. on Comp. Vis.*, pp. 35-42, Jan. 1998.

[8] Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura, "Visual Surveillance and Monitoring System Using an Omnidirectional Video Camera," *Proc. 14th Int'l. Conf. Patt. Recog.*, pp. 588-592, Aug. 1998.

[9] K. S. Huang, M. M. Trivedi, "Video Arrays for Real-Time Tracking of Persons, Head and Face in an Intelligent Room," *Machine Vision and Applications*, vol. 14, no. 2, pp. 103-111, June 2003.

[10] C. BenAbdelkader, P. Burlina, and L. Davis, "Single Camera Multiplexing for Multi-Target Tracking," *Proc. of IEE Int'l Conf. on Image Processing and its Application (IPA '99)*, pp. 1140-1143, 1999.

[11] W. Kruger, "Robust Real Time Ground Plane Motion Compensation from a Moving Vehicle," *MVA*, vol. 11, pp. 203-212, Nov. 1999.

[12] M. J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *CVIU*, vol. 63, no. 1, pp. 75-104, 1996.

[13] T. Gandhi, M. M. Trivedi, "Motion Analysis of Omni-Directional Video Streams for a Mobile Sentry," *1st ACM Int'l. Workshop on Video Surveillance,* Nov. 2003.

[14] B. Hall, K. Huang, M. M. Trivedi, "A Televiewing System for Multiple Simultaneous Customized Perspectives and Resolutions," *Proc. IEEE Int'l Conf. on Intelligent Transportation Systems*, Aug. 2001.